

containing a content entity that is identified by one of the content entity identifiers contained in the list.

DeRose relates to a data processing system for generating a representation of an electronic document and discloses structures and methods to facilitate navigating the electronic document and displaying it on an output device. Those electronic documents can include text content, descriptive mark-up and possibly non-text content. See col. 7, lines 59 - 62. Fig. 3 shows a representation of an electronic document, such as a book, having a hierarchical structure of chapters, sections, etc. Fig. 4 shows a representation of a mark-up of such a document that includes tags such as those used in a Standard Generalized Mark-up Language (SGML) document. See col. 8. DeRose shows in Fig. 6 a structure referred to as an element directory 91 that includes element descriptors 90. Each element descriptor corresponds to an element, such as element 70 in Fig. 5, of the electronic document. Each element descriptor includes a variety of information about the element such as its parent element (92), first and last children (94 and 96), left and right sisters (98 and 100), and a type name 102 that indicates the type of element in the document (i.e., book, title, etc.). The type name field 102 can include a pointer to a fully qualified name table, shown in Fig. 7, that contains a list of the types of elements (e.g., book, title, author, etc.) in the document. The element directory 91 also includes a location field 104 that contains a pointer to the location within a file where text characters for a text chunk are stored. See col. 9, lines 32 - 34. The text chunk is saved in an open text file and its location within that single text file is recorded in the location field 104 of the element descriptor 90. See col. 12, lines 41 - 44.

The Examiner asserts in the Office Action that the tree-like structure disclosed in DeRose, in Fig. 3, which has a plurality of chapters, sections and text, corresponds to the claimed

content object. The rejection is based on the position that DeRose's element directory 91, which includes a plurality of element descriptors 90 that point to text chunks in a text file, corresponds to the claimed file object. The Examiner agrees that DeRose does not disclose the text chunks being contained in a plurality of files, as the claims require, but asserts that it would have been obvious to modify the teachings of DeRose to store the text chunks in a plurality of files instead of the single text file DeRose discloses.

Applicant respectfully submits that the prior art does not suggest modifying the teachings of DeRose to include all the claim limitations, for at least the following reasons. Claim 1 recites a plurality of file objects, each containing a content entity. DeRose neither teaches nor suggests storing the text chunks in a plurality of files, but rather, discloses only that a text chunk is contained within a single text file. ("The text of the text chunk is then saved in the open text file and its location in the text file is recorded in the location field 104 of the element descriptor 90 for this text chunk." Col. 12, lines 41-44.) Although DeRose discloses as an alternative making the text chunks an integral part of the containing element, DeRose neither discloses nor suggests as an alternative storing the text chunks in a plurality of files.

It is respectfully submitted that since DeRose neither teaches nor suggests using a plurality of file objects to contain content entities, DeRose does not render claim 1 unpatentable.

The remaining independent claims, namely claims 8, 16, 23, 31 and 38 each also require a plurality of file objects, each containing a content entity. Accordingly, it is respectfully submitted that these claims are patentable for at least the same reasons discussed above.

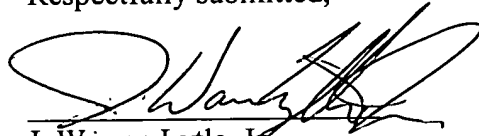
The remaining claims depend from one of the independent claims discussed above, and hence, are patentable for at least the same reasons.

New dependent claims 46 - 51 are added and specify that the content entity identifiers identify the content entities without specifying locations of the content entities. This limitation is supported throughout the written description. For example, the Product Structure Files (PSF) described beginning at page 34 show sequence identifiers that correspond to attributes of a book such as an ISBN number along with chapter and section indicators which do not specify a location.

In view of the foregoing, Applicants respectfully request the Examiner to find the application in condition for allowance. However, if for any reason the Examiner believes that the application is not now in condition for allowance, the Examiner is respectfully requested to call the undersigned to resolve any issues and to expedite the disposition of the application.

Applicant hereby petitions for any extension of time that may be required to maintain the pendency of this case, and any required fee for such extension is to be charged to Deposit Account No. 05-0460.

Respectfully submitted,



J. Warren Lytle, Jr.
Registration No. 39,283

EPSTEIN, EDELL, SHAPIRO, FINNAN & LYTLE, LLC
1901 Research Blvd., Suite 400
Rockville, Maryland 20850-3164
(301) 424-3640

Hand Delivered on: June 11, 2002

Version With Markings to Show Changes Made

Set forth below are the replacement paragraphs of the specification rewritten in the accompanying Amendment, marked up to show all changes relative to the previous version of those claims, in accordance with 37 C.F.R. §1.121(b)(1)(iii).

Amend the specification beginning at page 1, line 11 and ending at page 1, line 51, as follows.

A System and Method for Creating Compilations of Content

[__ / __, __] Serial No. 09/489,134 (Our reference Docket # STL000012US1)

Method and System for Adding Content to a Content Object Stored in a Data Repository

[__ / __, __] Serial No. 09/489,576 (Our reference Docket # STL000013US1)

Method and System for Adding User-Provided Content to a Content Object Stored in a Data Repository

[__ / __, __] Serial No. 09/488,976 (Our reference Docket # STL000014US1)

Method and System for Moving Content in a Content Object Stored in a Data Repository

[__ / __, __] Serial No. 09/488,971 (Our reference Docket # STL000015US1)

Method and System for Removing Content in a Content Object Stored in a Data Repository

[__ / __, __] Serial No. 09/489,087 (Our reference Docket # STL000016US1)

Prerequisite Checking in a System for Creating Compilations of Content

[__ / __, __] Serial No. 09/488,969 (Our reference Docket # STL000017US1)

Method and System for Preventing Mutually Exclusive Content Entities Stored in a Data Repository to be Included in the Same Compilation of Content

[__ / __, __] Serial No. 09/489,265 (Our reference Docket # STL000018US1)

Volume Management Method and System for a Compilation of Content

[__ / __, __] Serial No. 09/489,090 (Our reference Docket # STL000019US1)

Method and System for Calculating Cost of a Compilation of Content

[__ / __, __] Serial No. 09/489,143 (Our reference Docket # STL000020US1)

Method and System for Storing Hierarchical Content Objects in a Data Repository

[__ / __, __] Serial No. 09/489,570 (Our reference Docket # STL000021US1)

Providing a Functional Layer for Facilitating Creation and Manipulation of Compilations of Content

Serial No. [] / [], [] 09/489,605 (Our reference Docket # STL000023US1)

A Hitmask for Querying Hierarchically Related Content Entities

[] / [], [] Serial No. 09/489,133 (Our reference Docket # STL990182US1)

A Method and Configurable Model for Storing Hierarchical Data in a Non-Hierarchical Data Repository

[] / [], [] Serial No. 09/489,561 (Our reference Docket # STL000025US1)

Reference to a Computer Listing Appendix

Appendix A to this application is set forth on a single compact and the material recorded thereon is incorporated by reference herein. The following file is recorded on the compact disc: file name: AppendixA.txt; file size: 107kB; date of creation: May 16, 2002.--

Amend the paragraph beginning at page 6, line 6, as follows.

Figs. 22A – [22D] 22E represent the system administrator interface of an embodiment of the present invention;

Amend the paragraph beginning at page 6, line 10, as follows.

Fig. [25] 24 is a state diagram representing the states of a user, request and CBO at various stages of the process for creating compilations of content.

Claims

Set forth below are the claims rewritten in the accompanying Amendment, marked up to show all changes relative to the previous version of those claims, in accordance with 37 C.F.R. §1.121(c)(ii).

1. (Amended) A file structure for storing a content object having a plurality of content entities, comprising:

a file object containing a list of content entity identifiers defining the content of the content object; and

a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.

8. (Amended) A file structure for storing a hierarchically structured content object having a plurality of content entities, comprising [the step of]:

a first file object containing an outline of containers and content entity identifiers defining the content and hierarchical structure of the content [entity] object; and

a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.

16. (Amended) A method for providing a file structure for storing a content object having a plurality of content entities, comprising the steps of:

creating a file object containing a list of content entity identifiers defining the content of the content object; and

creating a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.

23. (Amended) A method for storing a hierarchically structured content object having a plurality of content entities, comprising the steps of:

creating a first file object containing an outline of containers and content entity identifiers defining the content and hierarchical structure of the content [entity] object; and

creating a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.

31. (Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for providing a file structure for storing a content object having a plurality of content entities, comprising [the steps of]:

program instructions for creating a file object containing a list of content entity identifiers defining the content of the content object; and

program instructions for creating a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.

38. (Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for storing a hierarchically structured content object having a plurality of content entities, comprising [the steps of]:

program instructions for creating a first file object containing an outline of containers and content entity identifiers defining the content and hierarchical structure of the content [entity] object; and

program instructions for creating a plurality of file objects, each containing a content entity identified by one of the content entity identifiers contained in said list.